(0, 1)	y = = Bottom	y = y + 1 Bottom = Bottom + 1 AllocationDirection = (-1 ,0)	y = y + 1 Bottom = Bottom + 1 AllocationDirection = (1 ,0)

8.3.4.2 Allocation order for raster scan

For the raster scan slice group macroblock allocation map type, the first macroblock in the allocation order is the top-left one of the picture, and the allocation order follows the raster scan order.

For the reverse raster scan slice group macroblock allocation map type, the first macroblock in the allocation order is the bottom-right one of the picture, and the allocation order follows the reverse raster scan order.

8.3.4.3 Allocation order for wipe

For the wipe right slice group macroblock allocation map type, the first macroblock in the allocation order is the top-left one of the picture. The allocation order runs from top to bottom. The next macroblock after the bottom macroblock of a column is the top macroblock of the column to the right of the previous column.

For the wipe left slice group macroblock allocation map type, the first macroblock in the allocation order is the bottom-right one of the picture. The allocation order runs from bottom to top. The next macroblock after the top macroblock of a column is the bottom macroblock of the column to the left of the previous column.

8.3.4.4 Allocation order for macroblock level adaptive frame and field coding

Allocation order follows Figure 6-4 in subclause 6.2, instead of raster scan.

8.3.5 Data partitioning

When data partitioning is not used, coded slices start with a slice header and are followed by all the entropy coded symbols of Categories 4, 5, and 6 (see Category column in clause 7) of the macroblock data for the macroblocks of the slice.

When Data Partitioning is used, the macroblock data of a Slice is partitioned in three partitions. Partition A contains a partition A header and all entropy coded symbols of Category 4. Partition B contains a partition B header all symbols of Category 5. Partition C contains a partition C header and all symbols of Category 6. When data partitioning is used, each partition is conveyed in its own NAL unit, which may be empty if no symbols of the respective Category.

NOTE - Symbols of Category 5 are relevant to the decoding of intra coded texture information. Symbols of Category 6 are relevant to the decoding of residual data in Inter slices. Category 4 encompasses all other symbol types related to the decoding of macroblocks, and their information is often denoted as header information. The Partition A header contains all the symbols of the slice header, and additionally a slice number that is used to associate the partitions B and C with the partition A. The partition B and C headers contain only the slice number which allows their association with the partition A of the slice

8.3.6 Decoder process for management and use of the reference picture buffer

8.3.6.1 General

Formatted: Heading 4 Char Char1,Heading 4 Char1 Char Char,Heading 4 Char Char Char Char

Intro to multi picture buffer.

Decoder stores reference pictures as indicated in the bitstream. These are used for prediction. The buffer is divided into two independent buffers: the short term buffer and the long term buffer. Pictures can only remain in the short term buffer for a finite duration, given by MAX_FN. Pictures can remain in the long term buffer until the next IDR picture. mmco commands are used to control the contents of these buffers.

The decoder employs indices when referencing a picture for motion compensation on the macroblock layer. Default indices are defined. These indices of pictures in the reference picture buffer are re-mapped onto newly specified indices according to the remapping_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_idx fields.

8.3.6.2 Picture Numbering

Picture numbers are used in the decoding process for management and use of the reference picture buffer for both changing the default indices and for controlling the contents of the reference picture buffer using memory management control operations.

In frame structured pictures, the picture number, PN, of a frame that has frame number FN, is given by PN = FN

In field structured pictures, the picture number, PN, of a field that has frame number FN, is given by PN = 2 x FN if the field is a top field, and is given by PN = 2 x FN + 1 if the field is a bottom field.

Long term picture numbers are also used in the decoding process for management and use of the reference picture buffer. Long term picture numbers are used for both changing the default indices of pictures in the long term buffer, and are

used for transferring pictures from short term buffer to the long term buffer and for removing pictures from the long term buffer.

In frame structured pictures, the long term picture number, LTPN, of a frame that has long-term frame index LTFI, is given by LTPN = LTFI

In field structured pictures, the long term picture number, LTPN, of a field that has long-term frame index LTFI, is given by LTPN = 2 x LTFI if the field is a top field, and is given by LTPN = 2 x LTFI + 1 if the field is a bottom field.

In frame-structured pictures, the parameter MAX_PN is defined to equal MAX_FN, and in field-structured pictures , the parameter MAX_PN is defined to equal 2 x MAX_FN.

~~8.3.6 Decoder process for reference picture reordering operation~~

~~abs_diff_pic_num_minus1 plus one indicates the absolute difference between the frame_num of the frame being re-mapped and the prediction value. For the first occurence of the abs_diff_pic_num_minus1 field in ref_idx_reordering(), the prediction value is the frame_num of the current picture. For subsequent occurences of the abs_diff_pic_num_minus1 field in ref_idx_reordering(), the prediction value is the frame_num of the last frame that was re-mapped using abs_diff_pic_num_minus1.~~

~~The decoder shall determine the frame_num of the frame being re-mapped, FNQ, in a manner mathematically equivalent to the following, where the frame_num prediction is FNP.~~

```
if(remapping_of_pic_nums_idc = = 0) {       /* a negative difference */
  if(FNP – abs_diff_pic_num_minus1 < 0)
    FNQ = FNP – abs_diff_pic_num_minus1 + MAX_FN;
  else
    FNQ = FNP – abs_diff_pic_num_minus1;
}else{                          /* a positive difference */
  if(FNP + abs_diff_pic_num_minus1 > MAX_FN-1)
    FNQ  = FNP + abs_diff_pic_num_minus1 – MAX_FN;
  else
    FNQ  = FNP + abs_diff_pic_num_minus1;
}                                                              (8-9)
```

~~The encoder shall control remapping_of_pic_nums_idc and abs_diff_pic_num_minus1 such that the decoded value of abs_diff_pic_num_minus1 shall not be greater than or equal to MAX_FN.~~

~~As an example implementation, the encoder may use the following process to determine values of abs_diff_pic_num_minus1 and remapping_of_pic_nums_idc to specify a re-mapped frame number in question, FN:~~

```
DELTA = FNQ – FNP;
if(DELTA < 0) {
  if(DELTA < –MAX_PNF/2-1)
    MDELTA = DELTA + MAX_PNF;
  else
    MDELTA = DELTA;
}else{
  if(DELTA > MAX_PNF/2)
    MDELTA = DELTA – MAX_PNF;
  else
    MDELTA = DELTA;
}
abs_diff_pic_num_minus1 = abs(MDELTA);
DELTA = FNQ – FNP;
if(DELTA < 0) {
  if(DELTA < 1-(MAX_FN/2))
```

```
~~    MDELTA = DELTA + MAX_FN;~~
~~  else~~
~~    MDELTA = DELTA;~~
~~}else{~~
~~  if(DELTA > MAX_FN/2)~~
~~    MDELTA = DELTA - MAX_FN;~~
~~  else~~
~~    MDELTA = DELTA;~~
~~}~~
~~abs_diff_pic_num_minus1 = abs(MDELTA) - 1;~~
```

~~(8-10)~~

~~where abs( ) indicates an absolute value operation. remapping_of_pic_nums_idc is then determined by the sign of MDELTA.~~

~~The prediction value used by any subsequent abs_diff_pic_num_minus1 re-mappings is not affected by long_term_pic_idx.~~

8.3.6.~~4~~3 Default index orders

8.3.6.~~4~~3.1 General

A reference index is a relative index into a list of reference indices to indicate which reference picture out of the reference picture buffer is used for motion compensation. When decoding a P or SP slice, there is one such list of reference indices, called the first reference index list. When decoding a B ~~picture~~slice, there may be two reference indices used per block each pointing into a separate lists of reference indices which are called the first reference index list and second reference index list.

The first reference index list and the second reference index list have default mappings to the pictures numbers in the reference picture buffer as defined below. ~~The picture number is referred to as a frame number when decoding a frame or a field number when decoding a field. The field numbers of the reference fields are inferred from the frame numbers of the reference frames.~~

~~The syntax elements remapping_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_idx fields allow the order of the relative indexing into the reference picture buffer to be temporarily altered from the default index order for the decoding of a particular slice. An remapping_of_pic_nums_idc "end loop" indication indicates the end of a list of re-ordering commands for list 0 or list 1.~~

8.3.6.~~4~~3.2 Default index order for P and SP slices in frame-structured pictures

Formatted: Heading 5,h5,H5,H51,Titre 5

The default index order for list 0 prediction of P ~~frames~~ and SP slices in frame-structured pictures ~~is for the short-term frames~~ (i.e., frames which have not been given a long-term index) to precede the long-term frames in the reference indexing order. Within the set of short-term frames, the default order is for the frames to be ordered starting with the most recently ~~transmitted~~ decoded reference frame and proceeding through to the reference frame in the short-term buffer that was ~~transmitted~~ decoded first (i.e., in decreasing order of ~~frame number~~frame_num in the absence of wrapping of the ~~ten-bit frame number~~frame_num ~~field~~value). Within the set of long-term frames, the default order is for the frames to be ordered starting with the frame with the smallest long-term index and proceeding up to the frame with largest long-term index ~~equal to the most recent value of max_long_term_pic_idx_plus1-1~~.

A field that is stored in the short term or long term buffer for which the opposite parity field is not stored in the same buffer, are not included in the default index order, shall not be remapped, and shall not be used for prediction in frame-structured pictures.

A field that is stored in the short term or long term buffer for which the opposite parity field is not stored in the same buffer, are not included in the default index order, shall not be remapped, and shall not be used for prediction in frame-structured pictures.

For example, assuming no wrap of the frame_num field, if the buffer contains three short-term frames with frame_num equal to 300, 302, and 303 and two long-term frames with long-term frame indices 0 and 3, the default index order is:

default relative index 0 refers to the short-term frame with frame_num 303,

default relative index 1 refers to the short-term frame with frame_num 302,

default relative index 2 refers to the short-term frame with frame_num 300,

default relative index 3 refers to the long-term frame with long-term frame index 0, and

default relative index 4 refers to the long-term frame with long-term frame index 3.

~~For example, if the buffer contains three short-term frames with short-term frame numbers 300, 302, and 303 (which were transmitted in increasing frame number order) and two long-term frames with long-term frame indices 0 and 3, the default index order is:~~

~~default relative index 0 refers to the short-term frame with frame number 303,~~

~~default relative index 1 refers to the short-term frame with frame number 302,~~

~~default relative index 2 refers to the short-term frame with frame number 300,~~

~~default relative index 3 refers to the long-term frame with long-term frame index 0, and~~

~~default relative index 4 refers to the long-term frame with long-term frame index 3.~~

~~For the purposes of default reference picture index calculation, the transmitted order of reference frames is calculated as the order in which frames occurred in the bitstream, regardless of whether the transmitted frame had been coded as a pair of two separate field-structured pictures, or a single frame-structured picture.~~

8.3.6.3.3 Default index order for P and SP slices in field-structured pictures

In the case that the current picture is field-structured, each field of the stored reference pictures is identified as a separate reference picture with a unique index. Thus field structured pictures effectively have at least twice the number of pictures available for referencing. The calculated decoding order of reference fields alternates between reference pictures of the same and opposite parity, starting with fields that have the same parity as the current field-structured picture. Figure 8-6 shows the case of the first field in a field-structured picture pair, while Figure 8-7 shows the case of the second field. If one field of a reference frame was neither decoded nor stored, the decoding order calculation shall ignore the missing field and instead index the next available stored reference field of the respective parity in decoding order. When there are no more fields of the respective parity in the short term buffer, default indices shall be allocated to the not yet indexed fields of the other parity starting with the most recently decoded such field and progressing to the first decoded such field.

~~In the case that the current picture is field-structured, each field of the stored reference frames is identified as a separate reference picture with a unique index. Thus field-structured pictures effectively have twice the number of pictures available for referencing. The calculated transmission order of reference fields alternates between reference pictures of the same and opposite parity, starting with fields that have the same parity as the current field-structured picture. Figure 8-1 shows the case of the first field in a field-structured picture pair, while Figure 8-2 shows the case of the second field. If one field of a reference frame was neither transmitted nor stored, the transmission order calculation shall ignore the missing field and instead index the next available stored reference field of the respective parity in transmission order. The decoder shall treat the missing field as "unknown" data, while encoder shall not generate bitstreams that have data dependences on the missing field.~~

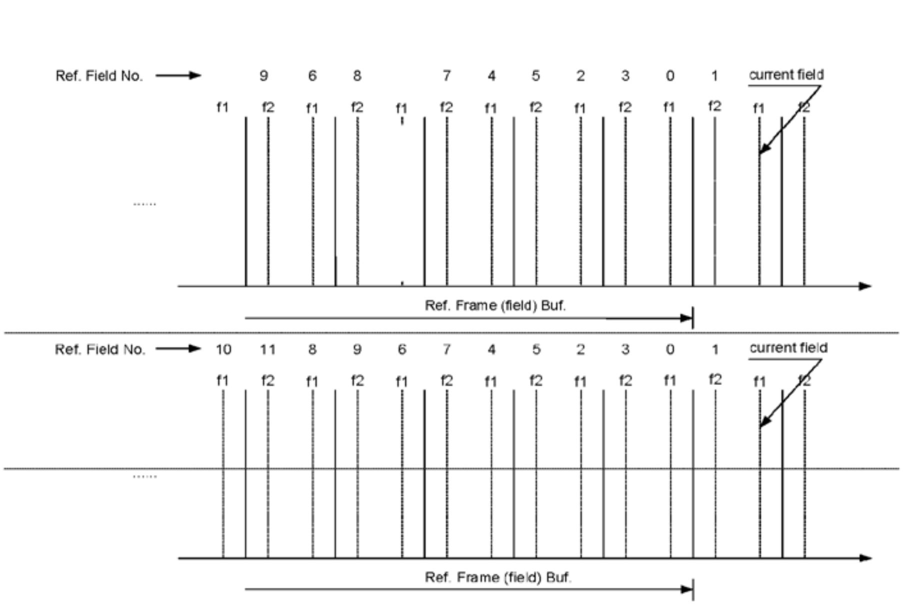

Figure 8-1 – Default reference field number assignment when the current picture is the first field coded in a frame

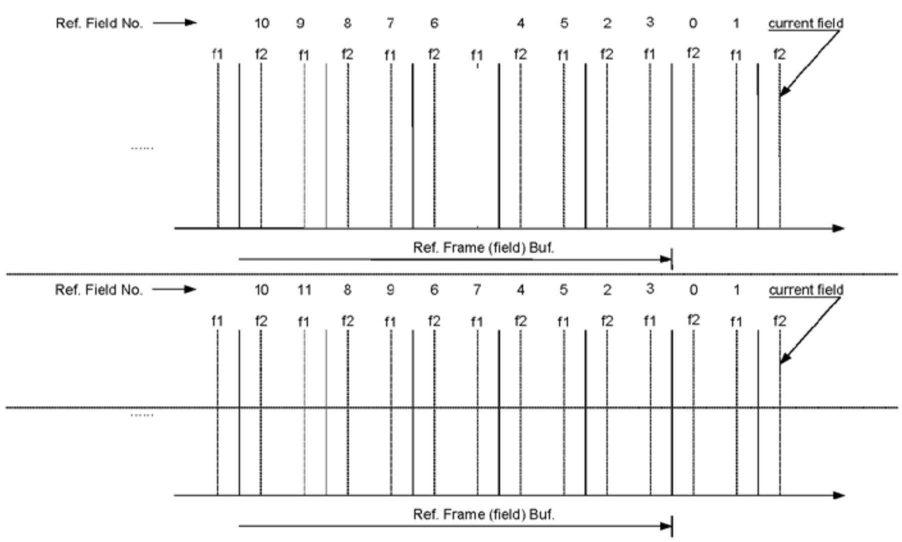

Figure 8-2 – Default reference field number assignment when the current picture is the second field coded in a frame

8.3.6.3.4 Default index order for B slices in frame-structured pictures

The organisation of short term pictures in the default order for B slices depends on output order, as given by PicOrderCnt.

The default index order for list 0 prediction of B slices in frame-structured pictures is for the short-term frames (i.e., frames which have not been given a long-term index) to precede the long-term frames in the reference indexing order. Within the set of short-term frames, the default order is for the frames to be ordered starting with the decoded reference frame with the largest value of PicOrderCnt less than the value of PicOrderCnt of the current frame and proceeding through to the reference frame in the short-term buffer that has the smallest value of PicOrderCnt; and then the frame with the largest value of PicOrderCnt greater than the value of PicOrderCnt of the current frame and proceeding through to the reference frame in the short-term buffer that has the smallest value of PicOrderCnt greater than the value of PicOrderCnt of the current frame. Within the set of long-term frames, the default order is for the frames to be ordered starting with the frame with the smallest long-term index and proceeding up to the frame with the largest long-term index.

The default index order for list 1 prediction of B slices in frame-structured pictures is for the short-term frames (i.e., frames which have not been given a long-term index) to precede the long-term frames in the reference indexing order. Within the set of short-term frames, the default order is for the frames to be ordered starting with the decoded reference frame with the largest value of PicOrderCnt and proceeding through to the reference frame in the short-term buffer that has the smallest value of PicOrderCnt. Within the set of long-term frames, the default order is for the frames to be ordered starting with the frame with the smallest long-term index and proceeding up to the frame with the largest long-term index.

The ordinary default order defined in the previous paragraph shall be used as the default index order for list 1 prediction unless there is more than one reference picture in the set and the ordinary default index order for list 1 prediction is the same as the default index order for list 0 prediction. In this exceptional case, the default index order for list 1 prediction shall be the ordinary default index order with the order of the first two pictures switched.

A field that is stored in the short term or long term buffer for which the opposite parity field is not stored in the same buffer, are not included in the default index order, shall not be remapped, and shall not be used for prediction in frame-structured pictures.

8.3.6.3.5 Default index order for B slices in field-structured pictures

The default index order for list 0 and list 1 prediction of B slices in field-structured pictures is as for frame-structured pictures except that it is split between even indices for same-parity fields and odd indices for opposite-parity fields.

8.3.6.4 Changing the default index orders

Formatted: Heading 5,h5,H5,H51,Titre 5

8.3.6.4.1 General

The syntax elements remapping_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_idx fields allow indexing into the reference picture buffer to be temporarily altered from the default index order for the decoding of the current slice. A remapping_of_pic_nums_idc "end loop" indication indicates the end of a list of re-ordering commands.

The indices are assigned starting at zero and increasing by one for each remapping_of_pic_nums_idc field. Pictures that are not re-mapped to a specific order by remapping_of_pic_nums_idc, shall follow after any pictures having a re-mapped order in the indexing scheme, following the default order amongst these non-re-mapped pictures.

8.3.6.4.2 Changing the default index orders for short term pictures

abs_diff_pic_num_minus1 plus one indicates the absolute difference between the picture number of the picture being re-mapped and the prediction value. For the first occurence of the abs_diff_pic_num_minus1 field in ref_idx_reordering(), the prediction value is the picture number of the current picture. For subsequent occurences of the abs_diff_pic_num_minus1 field in ref_idx_reordering(), the prediction value is the picture number of the picture that was re-mapped most recently using abs_diff_pic_num_minus1.

The decoder shall determine the picture number of the picture being re-mapped, PNQ, in a manner mathematically equivalent to the following, where the picture number prediction is PNP.

```
if(remapping_of_pic_nums_idc = = 0)
{   /* a negative difference */
  if(PNP – abs_diff_pic_num_minus1 < 0)
    PNQ = PNP – abs_diff_pic_num_minus1 + MAX_PN;
  else
    PNQ = PNP – abs_diff_pic_num_minus1;
}
else
{   /* a positive difference */
```

```
  if(PNP + abs_diff_pic_num_minus1 > MAX_PN-1)
    PNQ  = PNP + abs_diff_pic_num_minus1 - MAX_PN;
  else
    PNQ  = PNP + abs_diff_pic_num_minus1;
}
```

The encoder shall control remapping_of_pic_nums_idc and abs_diff_pic_num_minus1 such that the decoded value of abs_diff_pic_num_minus1 shall not be greater than or equal to MAX_PN.

As an example implementation, the encoder may use the following process to determine values of abs_diff_pic_num_minus1 and remapping_of_pic_nums_idc to specify a re-mapped picture number in question, PN:

```
if(remapping_of_pic_nums_idc = = 0)
{   /* a negative difference */
  if(PNP - abs_diff_pic_num_minus1 < 0)
    PNQ = PNP - abs_diff_pic_num_minus1 + MAX_PN;
  else
    PNQ = PNP - abs_diff_pic_num_minus1;
}
else
{   /* a positive difference */
  if(PNP + abs_diff_pic_num_minus1 > MAX_PN-1)
    PNQ  = PNP + abs_diff_pic_num_minus1 - MAX_PN;
  else
    PNQ  = PNP + abs_diff_pic_num_minus1;
}
```

where abs( ) indicates an absolute value operation.

remapping_of_pic_nums_idc is then determined by the sign of MDELTA.

The prediction value used by any subsequent abs_diff_pic_num_minus1 re-mappings is not affected by long_term_pic_idx.

8.3.6.4.3 Changing the default index orders for long term pictures

The long_term_pic_idx field indicates the long term picture number of the long term picture being remapped.

8.3.6.5 Overview of decoder process for reference picture buffer management

The reference picture buffer consists of two independent parts: a short term buffer and a long term buffer. The decoder shall assume the initial size of the long term buffer to be 0, that is, it shall assume that max_long_term_pic_idx_plus1 is set to zero.

The long term buffer has capacity to store max_long_term_idx_plus1 frames. The usage of the long term buffer is constrained so that it has capacity for no more than max_long_term_idx_plus1 top fields and no more than max_long_term_idx_plus1 bottom fields.

The remainder of the reference picture buffer is allocated to the short term buffer, which has capacity to store (num_of_ref_frames - max_long_term_idx_plus1) frames. There is no further constraint on its capacity to store top and bottom fields. For example, the whole of the short term buffer could be used to store top fields.

nal_storage_idc indicates whether the current picture is stored in the reference picture buffer. When nal_storage_idc is equal to 0, the current picture is not stored in the reference picture buffer, otherwise it is stored in the reference picture buffer.

If the current picture is stored in the reference picture buffer, the process used for storing is indicated by ref_pic_buffering_mode, which indicates either "Sliding Window", a first-in, first-out mechanism, or "Adaptive Memory Control", a customised adaptive buffering operation specified with memory_management_control_operation commands.

In frame structured pictures, memory_management_control_operation commands apply to both fields of the frame.

In field structured pictures, memory_management_control_operation commands apply to individual fields.

8.3.6.6 Sliding window reference picture buffer management

The "Sliding Window" buffering mode operates as follows.

If there is sufficient "unused" capacity in the short term buffer to store the current picture, the current picture is stored in the short term buffer and no pictures are removed from the short term buffer.

Otherwise if the current picture is a field-structured picture, the short-term field with the largest default index, that is, field that has been in the short term buffer for the longest time, is marked "unused", thus creating sufficient capacity to store the current picture. The current picture is then stored in the short term buffer.

Otherwise if the current picture is a frame-structured picture, default indices are calculated as done when decoding a field-structured picture, and the short-term field with the largest default index is marked "unused". If there is still insufficient "unused" capacity in the short term buffer to store the current picture, the short-term field which now has the largest default index is also marked "unused". The current picture is then stored in the short term buffer.

8.3.6.7 Adaptive Memory Control reference picture buffer management

Formatted: Heading 5,h5,H5,H51,Titre 5

8.3.6.7.1 General

The "Adaptive Memory Control" buffering mode allows specified pictures to be removed from either or both of the short and long term buffers, allows specified pictures to be moved from the short term buffer to the long term buffer, allows specified pictures to be removed from the long term buffer, allows the number of long term pictures to be modified, and allows the whole buffer to be reset, by use of memory_management_control_operation commands.

memory_management_control_operation commands are processed in the order they occur in the bitstream, and are processed after the whole picture has been decoded. When all commands have been processed, storage of the current picture is considered. When nal_storage_idc is equal to 0, the current picture is not stored in the reference picture buffer, otherwise it is stored in the short term buffer. memory_management_control_operation commands in the bitstream shall be such that when nal_storage_idc indicates that the current picture is to be stored, that there shall be sufficient "unused" capacity in the short term buffer to store the current picture.

8.3.6.7.2 Removal of short term pictures

If memory_management_control_operation equals 1 (Mark a Short-Term Picture as "Unused"), a specified short term picture in the short term buffer is marked as "unused", if that picture has not already been marked as "unused".

If the current decoded picture number is PNC, difference_of_pic_nums_minus1 is used in an operation mathematically equivalent to the following equations, to calculate, PNQ, the picture number of the short term picture to be marked as "unused".

```
if(PNC < difference_of_pic_nums_minus1)
  PNQ = PNC - difference_of_pic_nums_minus1 - 1 + MAX_PN;
else
  PNQ = PNC - difference_of_pic_nums_minus1 - 1;
```

Similarly, the encoder may compute the difference_of_pic_nums_minus1 value to encode using the following relation:

```
if(PNC < PNQ)
  difference_of_pic_nums_minus1 = PNC - PNQ - 1 + MAX_PN;
else
  difference_of_pic_nums_minus1 = PNC - PNQ - 1;
```

8.3.6.7.3 Removal of long term pictures

If memory_management_control_operation equals 2 (Mark a Long-Term Picture as "Unused"), a specified long term picture in the long term buffer is marked as "unused", if that picture has not already been marked as "unused".

The field long_term_pic_idx indicates the the long term picture number, LTPN, of the picture to be marked as "unused".

NOTE: this use of long_term_pic_idx is different to its use when transferring short term pictures to the long term buffer.

8.3.6.7.4 Transfer of short term pictures to the long term buffer

If memory_management_control_operation equals 3 (Assign a Long-Term Index to a Picture), a specified short term picture in the short term buffer is transferred to the long term buffer with a specified long-term index, if that picture has not already been transferred to the long term buffer. If the picture specified in a long-term assignment operation is already associated with the required long_term_pic_idx, no action shall be taken by the decoder. The specified short term

picture is no longer in the short term buffer following the processing of this command, and shall not be referenced at a later point in the bitstream by reference to its picture number.

If another picture is already present in the long term buffer with the same long-term index as the specified long-term index, the other picture is marked as "unused".

The picture in the short term buffer to be transferred is identified by its picture number, which is derived from difference_of_pic_nums_minus1 as in 9.2.6.6.1.

A top field in the short term buffer can only be transferred to the top field of a long term frame, and a bottom field in the short term buffer can only be transferred to the bottom field of a long term frame. The long term frame number of the frame into which the short term picture is transferred is given by long_term_pic_idx.

long_term_pic_idx shall not be greater than max_long_term_idx_plus1−1. If long_term_pic_idx does not satisfy this constraint, this condition should be treated by the decoder as an error.

For error resilience, the bitstream may contain the same long-term index assignment operation or max_long_term_idx_plus1 specification message repeatedly.

A bitstream shall not assign a long-term index to a short-term picture that has been marked as "unused" by the decoding process prior to the first such assignment message in the bitstream. A bitstream shall not assign a long-term index to a picture number that has not been sent.

Once a long-term picture index has been assigned to a picture, the only potential subsequent use of the long term picture's picture number within the bitstream shall be in a repetition of the long-term index assignment. long_term_pic_idx becomes the unique ID for the life of a long term picture.

8.3.6.7.5 Modification of the size of the long term buffer

If memory_management_control_operation equals 4 (Specify the Maximum Long-Term Frame Index), max_long_term_pic_idx_plus1 indicates the maximum index allowed for long-term reference frames (until receipt of another value of max_long_term_pic_idx_plus1).

If max_long_term_pic_idx_plus1 is smaller than its previous value, all frames in the long term buffer having indices greater than max_long_term_pic_idx_plus1 – 1 shall be marked "unused".

If max_long_term_pic_idx_plus1 is greater than its previous value, the capacity of the short term buffer is reduced by the same amount as the capacity of the long term buffer is increased. The memory_management_control_operation commands in the bitstream shall be such that at the time of processing this command, the reduced capacity of the short term buffer shall be sufficient for the contents of the short term buffer.

NOTE: max_long_term_pic_idx_plus1 can therefore be used to remove long term pictures from the long term buffer but can not be used to remove short term pictures from the short term buffer.

The frequency of transmitting max_long_term_idx_plus1 is out of the scope of this Recommendation. However, the encoder should send an max_long_term_idx_plus1 parameter upon receiving an error message, such as an Intra request message.

8.3.6.7.6 Buffer reset

If memory_management_control_operation equals 5 (Reset), or the current picture is an IDR picture, all pictures in the short and long term buffers are marked as "unused", and max_long_term_pic_idx_plus1 is set to zero.

8.3.6.8 Error resilience with reference picture buffer management

If required_frame_num_update_behaviour equals 1 the following picture buffer management behaviour shall be used.

If the decoder identifies that pictures that should have been stored have not been decoded, by a gap in frame numbers, the decoder shall act as if the missing pictures had been inserted into the reference picture buffer using the "Sliding Window" buffering mode. An index for a missing picture is called an "invalid" index. The decoder should infer an unintentional picture loss if any "invalid" index is referred to in motion compensation or if an "invalid" index is re-mapped.

If required_frame_num_update_behaviour equals 0, the decoder should infer an unintentional picture loss if one or several frame numbers are missing or if a picture not stored in the reference picture buffer is indicated in an abs_diff_pic_num_minus1 or long_term_pic_idx field.

Note: In case of an unintentional picture loss, the decoder may invoke some concealment process. If required_frame_num_update_behaviour equals 1, the decoder may replace the picture corresponding to an "invalid" index with an error-concealed one and remove the "invalid" indication. Otherwise, the decoder may insert an error-concealed picture into the reference picture buffer assuming the "Sliding Window" buffering mode. Concealment may be conducted by copying the closest temporally preceding picture that is available in the reference picture buffer into the position of the missing picture. The temporal

Formatted: Note 1

order of the short-term pictures in the reference picture buffer can be inferred from their picture numbers. In addition or instead, the decoder may send a forced intra update signal to the encoder by external means (for example, Recommendation H.245) if such external means is available, or the decoder may use external means or back-channel messages (for example, Recommendation H.245) to indicate the loss of pictures to the encoder if such external means is available.

~~8.3.6.1.2 Default index order for B pictures~~

~~The default index order for B frames is defined such that short-term frames that temporally precede the B frame are distinguished from short-term frames that temporally follow the B frame, based on the display order for each reference frame. The default order for list 0 prediction is specified in a manner that gives a lower index order to short-term frames that temporally precede the current frame, and the default order for list 1 prediction is specified in a manner that ordinarily gives a lower index order to short-term frames that temporally follow the current frame.~~

~~The default index order for B field pictures is split between even indices starting at 0 for fields of the same parity (top or bottom) as the current field and odd indices starting at 1 for fields of the opposite parity as the current field. This split ordering is analogous to the ordering defined in P pictures above except that a B field picture never references the opposite-parity field with which it shares the current frame.~~

~~8.3.6.1.2.1 List 0 prediction in B pictures~~

~~Within the set of short-term pictures, the default order for B frame list 0 prediction shall be for the frames to be ordered starting with the most recently-transmitted temporally-preceding reference frame and proceeding through to the temporally-preceding reference frame in the short-term buffer that was transmitted first. These temporally-preceding frames shall then be followed by the temporally-following reference frames, starting with the most recently-transmitted temporally-following reference frame in the short-term buffer and proceeding through to the temporally-following reference frame in the short-term buffer that was transmitted first. These frames shall then be followed by the long-term frames, starting with the frame with the smallest long-term index and proceeding up to the frame with long-term index equal to the most recent value of max_long_term_pic_idx_plus1-1. The default ordering for B field pictures follows in like manner except that it is split between even indices for same-parity fields and odd indices for opposite-parity fields.~~

~~8.3.6.1.2.2 List 1 prediction in B pictures~~

~~The default order for B picture list 1 prediction is defined in a similar manner as for list 0 prediction, but giving preference in the order for pictures that temporally follow the B picture and swapping the order of the first two pictures if this would result in an identical list 0 and list 1 default indexing order.~~

~~Within the set of short-term pictures, the ordinary default order for B picture list 1 prediction shall be for the pictures to be ordered starting with the most recently-transmitted temporally-following reference picture and proceeding through to the temporally-following reference picture that has been in the short-term buffer the longest. These temporally-following pictures shall then be followed by the temporally-preceding reference pictures, starting with the most recently-transmitted temporally-preceding reference picture and proceeding through to the temporally-preceding reference picture that has been in the short-term buffer the longest. These pictures shall then be followed by the long-term pictures, starting with the picture with the smallest long-term index and proceeding up to the picture with long-term index equal to the most recent value of max_long_term_pic_idx_plus1-1.~~

~~The ordinary default order defined in the previous paragraph shall be used as the default index order for list 1 prediction unless there is more than one reference picture in the set and the ordinary default index order for list 1 prediction is the same as the default index order for list 0 prediction. In this exceptional case, the default index order for list 1 prediction shall be the ordinary default index order with the order of the first two pictures switched.~~

~~8.3.6.1.3 Reordering of list 0 and list 1~~

~~The first abs_diff_pic_num_minus1 or long_term_pic_idx field that is received (if any) moves a specified picture out of the default order to the relative index of zero. The second such field moves a specified picture to the relative index of one, etc. The set of remaining pictures not moved to the front of the relative indexing order in this manner shall retain their default order amongst themselves and shall follow the pictures that have been moved to the front of the buffer in relative indexing order. Note that if the current picture is a frame, then abs_diff_pic_num_minus1 or long_term_pic_idx refer to the relative frame indices defined above; if the current picture is a field, then abs_diff_pic_num_minus1 or long_term_pic_idx refer to the relative field indices defined above.~~

~~If there is not more than one list 0 reference picture used, no more than one abs_diff_pic_num_minus1 or long_term_pic_idx field shall be present in the same RPS layer unless the current picture is a B picture. If the current picture is a B picture and there is not more than one list 1 reference picture used, no more than two abs_diff_pic_num_minus1 or long_term_pic_idx fields shall be present in the same RPS layer.~~

~~Any re-mapping of frame numbers specified for some slice shall not affect the decoding process for any other slice.~~

~~In a P picture an remapping_of_pic_nums_idc "end loop" indication is followed by ref_pic_buffering_mode. In a B picture, the first remapping_of_pic_nums_idc "end loop" indication, which concludes the remapping of the list 0~~

~~reference set, is followed by an additional remapping_of_pic_nums_idc indicator that begins the remapping operations (if any) for the list 1 reference set.~~

~~Within one RPS layer, remapping_of_pic_nums_idc shall not specify the placement of any individual reference picture into more than one re-mapped position in relative index order.~~

~~8.3.7 Decoder process for reference picture buffer management~~

~~In the "Sliding Window" buffering mode, the current decoded picture shall be added to the buffer with default relative index 0, and any marking of pictures as "unused" in the buffer is performed automatically in a first-in-first-out fashion among the set of short-term pictures. In this case, if the buffer has sufficient "unused" capacity to store the current picture, no additional pictures shall be marked as "unused" in the buffer. If the buffer does not have sufficient "unused" capacity to store the current picture, the picture with the largest default relative index among the short-term pictures in the buffer shall be marked as "unused". If in this case the current picture is the first field of a frame, then only the field of the same parity (top or bottom) will be marked as unused in the buffer, so that the second field of the current frame may still reference the other field of the largest relative index. In the sliding window buffering mode, no additional information is transmitted to control the buffer contents.~~

~~In the "Adaptive Memory Control" buffering mode, the encoder explicitly specifies any addition to the buffer or marking of data as "unused" in the buffer, and may also assign long-term indices to short-term frames. The current frame and other frames may be explicitly marked as "unused" in the buffer, as specified by the encoder. This buffering mode requires further information that is controlled by memory management control operation (memory_management_control_operation) parameters.~~

~~The decoder stores the reference pictures for inter-picture decoding in a reference picture buffer. The decoder replicates the reference picture buffer of the encoder according to the reference picture buffering mode and any memory management control operations specified in the bitstream. The buffering scheme may also be operated when partially erroneous pictures are decoded.~~

~~If the current decoded frame number is PNC and the decoded UVLC code number is difference_of_pic_nums, an operation mathematically equivalent to Equation 8-11 shall be used for calculation of PNQ, the specified frame number in question:~~

```
if(PNC – difference_of_pic_nums < 0)
  PNQ = PNC – difference_of_pic_nums + MAX_PN;
else
  PNQ = PNC – difference_of_pic_nums;
```

~~(8-11)~~

~~Similarly, the encoder may compute the difference_of_pic_nums value to encode using Equation 8-12:~~

```
if(PNC – PNQ < 0)
  difference_of_pic_nums = PNC – PNQ + MAX_PN;
else
  difference_of_pic_nums = PNC – PNQ;
```

~~(8-12)~~

~~For example, if the decoded value of difference_of_pic_nums is zero and memory_management_control_operation indicates marking a short-term frame as "unused", the current decoded frame shall be marked as "unused" (thus indicating that the current frame is not a stored frame).~~

~~Each coded and stored picture is assigned a Frame number (PN) which is stored with the picture in the reference picture buffer. PN represents a sequential picture counting identifier for stored pictures. PN is constrained, using modulo MAX_PN arithmetic operation. For the first coded picture of an IDR picture, PN shall be "0". For each and every other coded and stored picture, PN shall be increased by 1.~~

~~Besides the PN, each picture stored in the reference picture buffer has an associated index, called the default relative index. When a picture is first added to the reference picture buffer it is given default relative index 0 – unless it is assigned to a long-term index. The default relative indices of pictures in the reference picture buffer are modified when pictures are added to or removed from the reference picture buffer, or when short-term pictures are assigned to long-term indices.~~

~~The pictures stored in the reference picture buffers can also be divided into two categories: long-term pictures and short-term pictures. A long-term picture can stay in the reference picture buffer for a long time (more than MAX_PN-1 coded~~

~~and stored picture intervals). The current picture is initially considered a short-term picture. Any short-term picture can be changed to a long-term picture by assigning it a long-term index according to information in the bitstream. The PN is the unique ID for all short-term pictures in the reference picture buffer. When a short-term picture is changed to a long-term picture, it is also assigned a long-term picture index (long_term_pic_idx). A long-term picture index is assigned to a picture by associating its PN to an long_term_pic_idx. Once a long-term picture index has been assigned to a picture, the only potential subsequent use of the long-term picture's PN within the bitstream shall be in a repetition of the long-term index assignment. long_term_pic_idx becomes the unique ID for the life of a long-term picture.~~

~~PN (for a short-term picture) or long_term_pic_idx (for a long-term picture) can be used to re-map the pictures into re-mapped relative indices for efficient reference picture addressing.~~

~~8.3.7.1 Decoder process for short/long-term picture management~~

~~The decoder may have both long-term pictures and short-term pictures in its reference picture buffer. The max_long_term_idx_plus1 field is used to indicate the maximum long-term picture index allowed in the buffer. If no prior value of max_long_term_idx_plus1 has been sent, no long-term pictures shall be in use, i.e. max_long_term_idx_plus1 shall initially have an implied value of "0". Upon receiving an max_long_term_idx_plus1 parameter, a new max_long_term_idx_plus1 shall take effect until another value of max_long_term_idx_plus1 is received. Upon receiving a new max_long_term_idx_plus1 parameter in the bitstream, all long-term pictures with associated long-term indices greater than or equal to max_long_term_idx_plus1 shall be considered marked "unused". The frequency of transmitting max_long_term_idx_plus1 is out of the scope of this Recommendation. However, the encoder should send an max_long_term_idx_plus1 parameter upon receiving an error message, such as an Intra request message.~~

~~A short-term picture can be changed to a long-term picture by using an memory_management_control_operation command with an associated difference_of_pic_nums and long_term_pic_idx. The short-term frame number is derived from difference_of_pic_nums and the long-term picture index is long_term_pic_idx. Upon receiving such an memory_management_control_operation command, the decoder shall change the short-term picture with PN indicated by difference_of_pic_nums to a long-term picture and shall assign it to the long-term index indicated by long_term_pic_idx. If a long-term picture with the same long-term index already exists in the buffer, the previously-existing long-term picture shall be marked "unused". An encoder shall not assign a long-term index greater than max_long_term_idx_plus1-1 to any picture. If long_term_pic_idx is greater than max_long_term_idx_plus1-1, this condition should be treated by the decoder as an error. For error resilience, the encoder may send the same long-term index assignment operation or max_long_term_idx_plus1 specification message repeatedly. If the picture specified in a long-term assignment operation is already associated with the required long_term_pic_idx, no action shall be taken by the decoder. An encoder shall not assign the same picture to more than one long-term index value. If the picture specified in a long-term index assignment operation is already associated with a different long-term index, this condition should be treated as an error. An encoder shall only change a short-term picture to a long-term picture if its PN has not been used in any subsequent coded picture. An encoder shall not assign a long-term index to a short-term picture that has been marked as "unused" by the decoding process prior to the first such assignment message in the bitstream. An encoder shall not assign a long-term index to a frame number that has not been sent.~~

~~8.3.7.2 Decoder process for reference picture buffer mapping~~

~~The decoder employs indices when referencing a picture for motion compensation on the macroblock layer. In pictures other than B pictures, these indices are the default relative indices of pictures in the reference picture buffer when the fields abs_diff_pic_num_minus1 and long_term_pic_idx are not present in the current slice layer as applicable, and are re-mapped relative indices when these fields are present. In B pictures, the first one or two pictures (depending on BTPSM) in relative index order are used for list 1 prediction, and the list 0 picture reference parameters specify a relative index into the remaining pictures for use in list 0 prediction. [Ed. Note: needs to be changed]~~

~~The indices of pictures in the reference picture buffer can be re-mapped onto newly specified indices by transmitting the remapping_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_idx fields. remapping_of_pic_nums_idc indicates whether abs_diff_pic_num_minus1 or long_term_pic_idx is present. If abs_diff_pic_num_minus1 is present, remapping_of_pic_nums_idc specifies the sign of the difference to be added to a frame number prediction value. The abs_diff_pic_num_minus1 value corresponds to the absolute difference between the PN of the picture to be re-mapped and a prediction of that PN value. The first transmitted abs_diff_pic_num_minus1 is computed as the absolute difference between the PN of the current picture and the PN of the picture to be re-mapped. The next transmitted abs_diff_pic_num_minus1 field represents the difference between the PN of the previous picture that was re-mapped using abs_diff_pic_num_minus1 and that of another picture to be re-mapped. The process continues until all necessary re-mapping is complete. The presence of re-mappings specified using long_term_pic_idx does not affect the prediction value for subsequent re-mappings using abs_diff_pic_num_minus1. If remapping_of_pic_nums_idc indicates the presence of an long_term_pic_idx field, the re-mapped picture corresponds to a long-term picture with a long-term index of long_term_pic_idx. If any pictures are not re-mapped to a specific order by remapping_of_pic_nums_idc, these remaining pictures shall follow after any pictures having a re-mapped order in the indexing scheme, following the default order amongst these non-re-mapped pictures.~~

~~If the indicated sequence parameter set in the latest received slice or data partition signals the required frame number update behaviour, the decoder shall operate as follows. The default picture index order shall be updated as if pictures corresponding to missing frame numbers were inserted to the reference picture buffer using the "Sliding Window" buffering mode. An index corresponding to a missing frame number is called an "invalid" index. The decoder should infer an unintentional picture loss if any "invalid" index is referred to in motion compensation or if an "invalid" index is re-mapped.~~

~~If the indicated sequence parameter set in the latest received slice or data partition does not signal the required frame number update behaviour, the decoder should infer an unintentional picture loss if one or several frame numbers are missing or if a picture not stored in the reference picture buffer is indicated in a transmitted abs_diff_pic_num_minus1 or long_term_pic_idx.~~

> ~~Note: In case of an unintentional picture loss, the decoder may invoke some concealment process. If the required frame number update behaviour was indicated, the decoder may replace the picture corresponding to an "invalid" index with an error-concealed one and remove the "invalid" indication. If the required frame number update behaviour was not indicated, the decoder may insert an error concealed picture into the reference picture buffer assuming the "Sliding Window" buffering mode. Concealment may be conducted by copying the closest temporally preceding picture that is available in the reference picture buffer into the position of the missing picture. The temporal order of the short-term pictures in the reference picture buffer can be inferred from their default relative index order and PN fields. In addition or instead, the decoder may send a forced intra update signal to the encoder by external means if such means is available (for example, Recommendation H.245), or the decoder may use external means or back-channel messages (for example, Recommendation H.245) to indicate the loss of pictures to the encoder.~~

~~8.3.7.3 Decoder process for multi-picture motion compensation~~

~~Multi-picture motion compensation is applied if the use of more than one reference picture is indicated. For multi-frame motion compensation, the decoder chooses a reference picture as indicated using the reference frame fields on the macroblock layer. Once, the reference picture is specified, the decoding process for motion compensation proceeds.~~

~~8.3.7.4 Decoder process for reference picture buffering~~

~~The buffering of the currently decoded picture can be specified using the reference picture buffering mode (ref_pic_buffering_mode). The buffering may follow a first-in, first-out ("Sliding Window") mode. Alternatively, the buffering may follow a customized adaptive buffering ("Adaptive Memory Control") operation that is specified by the encoder.~~

~~The "Sliding Window" buffering mode operates as follows. First, the decoder determines whether the picture can be stored into "unused" buffer capacity. If there is insufficient "unused" buffer capacity, the short-term picture with the largest default relative index (i.e. the oldest short term picture in the buffer) shall be marked as "unused". The current picture is stored in the buffer and assigned a default relative index of zero. The default relative index of all other short-term pictures is incremented by one. The default relative indices of all long-term pictures are incremented by one minus the number of short-term pictures removed.~~

~~In the "Adaptive Memory Control" buffering mode, specified pictures may be removed from the reference picture buffer explicitly. The currently decoded picture, which is initially considered a short-term picture, may be inserted into the buffer with default relative index 0, may be assigned to a long-term index, or may be marked as "unused" by the encoder. Other short-term pictures may also be assigned to long-term indices. The buffering process shall operate in a manner functionally equivalent to the following: First, the current picture is added to the reference picture buffer with default relative index 0, and the default relative indices of all other pictures are incremented by one. Then, the memory_management_control_operation commands are processed:~~

~~If memory_management_control_operation indicates a reset of the buffer contents or if the current picture is the first one in an IDR picture, all pictures in the buffer are marked as "unused" except the current picture (which will be the picture with default relative index 0). Moreover, the maximum long-term index shall be reset to zero.~~

~~If memory_management_control_operation indicates a maximum long-term index using max_long_term_idx_plus1, all long-term pictures having long-term indices greater than or equal to max_long_term_idx_plus1 are marked as "unused" and the default relative index order of the remaining pictures are not affected.~~

~~If memory_management_control_operation indicates that a picture is to be marked as "unused" in the reference picture buffer and if that picture has not already been marked as "unused", the specified picture is marked as "unused" in the reference picture buffer and the default relative indices of all subsequent pictures in default order are decremented by one.~~

~~If memory_management_control_operation indicates the assignment of a long-term index to a specified short-term picture and if the specified long-term index has not already been assigned to the specified short-term picture, the specified short-term picture is marked in the buffer as a long-term picture with the specified long-term index. If another picture is already present in the buffer with the same long-term index as the specified long-term index, the other picture is marked as "unused". All short-term pictures that were subsequent to the specified short-term picture in default relative index order and all long-term pictures having a long-term index less than the specified long-term index have their~~

~~associated default relative indices decremented by one. The specified picture is assigned to a default relative index of one plus the highest of the incremented default relative indices, or zero if there are no such incremented indices.~~

8.3.6.9 Decoding process for macroblock level frame/field adaptive coding

When mb_ frame_field_adaptive_flag = = 1, the decoded frame is scanned on a macroblock pair by macroblock pair basis, as shown in Figure 6-4 (subclause 6.2). A macroblock pair can be decoded in either frame or field decoding mode. For frame decoding mode, a macroblock pair is decoded as two frame macroblocks, and each can be further divided into one of block patterns shown in Figure 6-4. For field coding mode, a macroblock pair is first split into one top-field macroblock and one bottom-field macroblock, as shown in Figure 8-3. The top-field macroblock and the bottom-field macroblock are further divided into block patterns shown in Figure 6-5. Each macroblock in either frame or field decoding mode can have a different mb_type described in subclause 7.4.6.

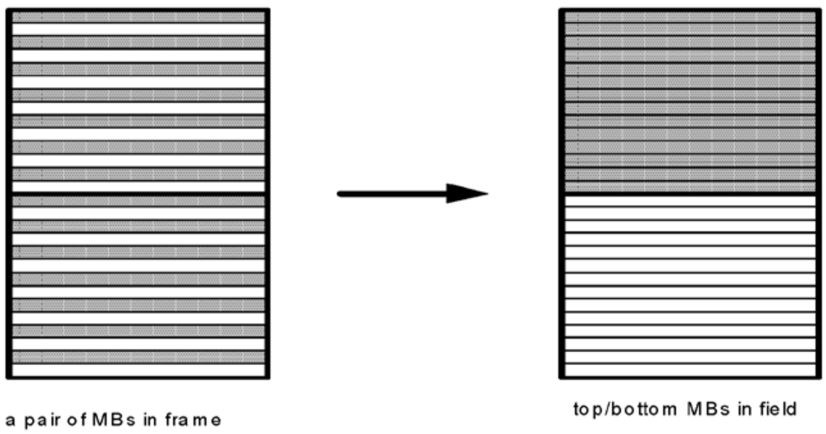

Figure 8-3 Split of a pair of macroblocks into one top-field macroblock and one bottom-field macroblock.

When mb_field_decoding_flag = = 0, the top macroblock of a macroblock pair is decoded first, followed by the bottom macroblock, as shown in Figure 6-4 (subclause 6.2). When mb_field_decoding_flag = = 1, the top-field macroblock is decoded first, followed by the bottom-field macroblock (see Figure 6-4). A few specific rules/conventions are specified as follows.

For intra prediction, if a block/macroblock is in field decoding mode, its neighbouring samples in calculating the prediction shall be the neighbouring samples of the same field.

As in frame decoding mode, the prediction mode of a 4x4 field block is decoded based upon the prediction modes of the (above and left) neighbouring blocks. For interior blocks of a field macroblock pair, the neighbouring blocks used in coding of intra prediction mode are the blocks above and left of the current block. For boundary blocks of a field macroblock pair, the above or left neighbouring block may be in different macroblock pair that can be of either frame or field coding mode. The neighbouring blocks for these boundary blocks shall be as follows:

– If the above or the left macroblock pair is also in field decoding mode, the neighbours of the boundary blocks in the current macroblock pair are in the same field of the above or the left macroblock pair.
– If the above or the left macroblock pair is in frame decoding mode, the neighbours of the boundary blocks in the top (bottom) field macroblock are defined to be the corresponding blocks in the top (bottom) macroblock in the frame macroblock pair.
– For macroblock pairs on the upper boundary of a slice, if the left macroblock pair is in frame decoding mode, then the intra mode prediction value used to predict a field macroblock shall be set to DC prediction.

8.4 Motion compensation

The motion compensation process generates motion compensated predictions for picture blocks using previously decoded reference pictures. The reference picture selection is described in subclauses 7.3.3, 7.3.5.1-2, 7.4.6.1, and 8.3.6-

7. If pic_structure indicates a field picture, only the reference field indicated by the ref_idx_l0 or ref_idx_l1 is used in the motion compensation. The motion vectors to be used are described in subclauses 7.3.5.1-2, 7.4.5, 7.4.6.1 and 8.4.1.

If the current MB pair is in frame mode and one or more neighbouring blocks is in field mode, the reference frame index used in MV prediction from the field coded neighbours is obtained by dividing the reference field list index by 2 and truncation any fractional result toward zero to obtain the effective frame index for prediction. If the MB pair is in frame mode, the reference field number for any frame coded neighbour is obtained by multiplying the reference frame index by 2.

When a macroblock pair is in field mode, each field macroblock may refer to any (top or bottom) field in the reference picture buffer.

8.4.1 Prediction of vector components

No vector component prediction takes place across macroblock boundaries of macroblocks that do not belong to the same slice. For the purpose of vector component prediction, macroblocks that do not belong to the same slice are treated as outside the picture.

With exception of the 16x8 and 8x16 block shapes, "median prediction" (see subclause 8.4.1.1) is used. In case the macroblock may be classified to have directional segmentation the prediction is defined in subclause 8.4.1.2. Motion vector for a Skip mode macroblock shall be obtained as described in subclause 8.4.1.3.

8.4.1.1 Median prediction

The prediction of the components of the motion vector value for a block E is formed based on the parameters of neighbouring blocks A, B, C, and D as shown in Figure 8-4. This process is referred to as median prediction.

A The block containing the sample to the left of the upper left sample in E

B The block containing the sample just above the upper left sample in E

C The block containing the sample above and to the right of the upper right sample in E

D The block containing the sample above and to the left of the upper left sample in E

NOTE - The prediction of A, B, C, D and E may use different indices into the reference picture list.

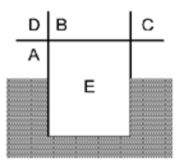

Figure 8-4 – Median prediction of motion vectors

The following rules specify the predicted motion vector value resulting from the median prediction process for block E:

- If block C is outside the current picture or slice or is not available due to the decoding order within a macroblock as specified in Figure 6-4, its motion vector and reference picture index shall be considered equal to the motion vector and reference picture index for block D.
- If blocks B, C, and D are all outside the current picture or slice, their motion vector values and reference picture indices shall be considered as equal to the motion vector value and reference picture index for block A.
- If any predictor not specified by the first or second rules above is coded as intra or is outside the current picture or slice, its motion vector value shall be considered equal to zero and it shall be considered to have a different reference picture than block E.
- If only one of the three blocks A, B and C has the same reference picture as block E, then the predicted motion vector for block E shall be equal to the motion vector of the A, B, or C block with the same reference picture as block E; otherwise, each component of the predicted motion vector value for block E shall be the median of the corresponding motion vector component values for blocks A, B, and C.

The following additional considerations apply in the case of macroblock-adaptive frame/field coding:

- If a block A is field type then it is assigned two frame MV's for the purpose of motion vector prediction. The first frame MV is the field MV of the block with vertical motion vector component multiplied by 2, and the second MV is the field MV of the block in same geometric location as A in the second MB of the MB pair (vertical motion vector component multiplied by two). If a block A is frame type then it is assigned two field MV's for the purpose of motion vector prediction. The first field MV is the frame MV of the block with vertical motion vector component divided by 2, and the second MV is the frame MV of the block in same geometric location as A in the second MB of the MB pair (vertical motion vector component divided by two). Similar rules are used to determine the two reference frames (fields) of a field (frame) block.
- If E is in frame coding mode, the MVs of A, B, C and D used in calculating PMV are also frame-based. If block A, B, C, or D is coded in field coding mode, its two frame-based MVs are averaged. In that case, the two reference field numbers of A, B, C or D shall be the same, and they shall be equal to the reference frame number of E multplied by 2.
- If E is in field coding mode, the MVs of A, B, C and D used in calculating PMV are also field-based in the same field parity. If block A, B, C, or D is frame coded, the field-based motion vector is obtained by averaging the two field MVs of the block. In that case, two frame reference numbers of A, B, C or D shall be the same, and they shall be equal to the reference field number of E divided by 2 with truncation of fractional values toward zero.

8.4.1.2 Directional segmentation prediction

If the macroblock where the block to be predicted is coded in 16x8 or 8x16 mode, the prediction is generated as follows (refer to Figure 8-5 and the definitions of A, B, C, E above):

a) Vector block size 8x16:
   1) Left block: A is used as prediction if it has the same reference picture as E, otherwise "median prediction" is used
   2) Right block: C is used as prediction if it has the same reference picture as E, otherwise "median prediction" is used
b) Vector block size 16x8:
   1) Upper block: B is used as prediction if it has the same reference picture as E, otherwise "median prediction" is used
   2) Lower block: A is used as prediction if it has the same reference picture as E, otherwise "median prediction" is used

If the indicated prediction block is outside the picture, the same substitution rules are applied as in the case of median prediction.

For field-coded macroblocks, the directional segmentation follow the same conventions as the above, but the neighbouring blocks are constructed from samples of the macroblock pair having the same field parity.

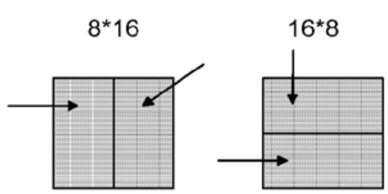

Figure 8-5 – Directional segmentation prediction

8.4.1.3 Motion vector for a skip mode macroblock

Motion vector for a Skip mode macroblock shall be obtained identically to the prediction motion vector for the 16x16 macroblock type. However, if any of the conditions below hold, a zero motion vector shall be used instead:

a) The Macroblock immediately above or to the left is not available (that is, is outside of the picture or belongs to a different slice)

b) Either one of the motion vectors applying to samples A or B (as described in subclause 8.4.1.1) uses the last decoded picture as reference and has zero magnitude.

8.4.1.4 Chroma vectors

Chroma vectors are derived from the luma vectors. Since chroma has half resolution compared to luma, the chroma vectors are obtained by dividing the corresponding luma motion vectors by two.

Due to the lower resolution of the chroma array relative to the luma array, a chroma vector applies to 1/4 as many samples as the luma vector. For example if the luma vector applies to 8x16 luma samples, the corresponding chroma vector applies to 4x8 chroma samples and if the luma vector applies to 4x4 luma samples, the corresponding chroma vector applies to 2x2 chroma samples.

8.4.2 Fractional sample accuracy

Fractional sample accuracy is indicated by motion_resolution. If motion_resolution has the value 0, quarter-sample interpolation with a 6-tap filter is applied to the luma samples in the block. If motion_resolution has the value 1, eighth-sample interpolation with an 8-tap filter is used. The prediction process for chroma samples in both cases is described in subclause 8.4.2.3.

8.4.2.1 Quarter sample luma interpolation

In Figure 8-6, the positions labelled with upper-case letters within shaded blocks represent reference picture samples at integer sample positions, and the positions labelled with lower-case letters within un-shaded blocks represent reference picture samples at fractional sample positions.

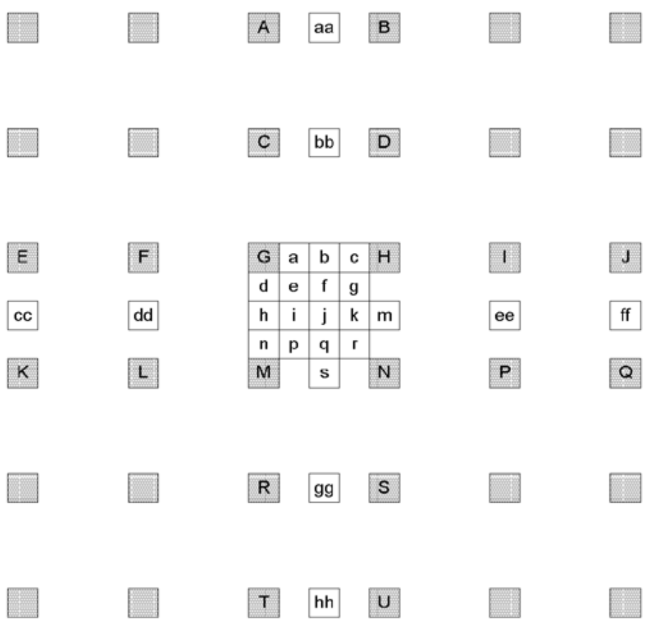

Figure 8-6 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation.

The luma prediction values at half sample positions shall be obtained by applying a 6-tap filter with tap values (1, -5, 20, 20, -5, 1). The luma prediction values at quarter sample positions shall be obtained by averaging samples at integer and half sample positions. The process for each fractional position is described below.

– The samples at half sample positions labelled 'b' shall be obtained by first calculating intermediate values denoted as '*b*' by applying the 6-tap filter to the nearest integer position samples in the horizontal

direction. The samples at half sample positions labelled 'h' shall be obtained by first calculating intermediate values denoted as '*h*' by applying the 6-tap filter to the nearest integer position samples in the vertical direction:

b=(E-5F+20G+20H-5I+J),

h=(A-5C+20G+20M-5R+T).

The final prediction values shall be calculated using:

b=Clip1((*b*+16)>>5),

h= Clip1((*h*+16)>>5).

– The samples at half sample position labelled as 'j' shall be obtained by first calculating intermediate value denoted as '*j*' by applying the 6-tap filter to the intermediate values of the closest half sample positions in either the horizontal or vertical direction because these yield an equivalent result.

j=*cc*-5*dd*+20*h*+20*m*-5*ee*+*ff*, or

j=*aa*-5*bb*+20*b*+20*s*-5*gg*+*hh*,

where intermediate values denoted as '*aa*', '*bb*', '*gg*', '*s*' and '*hh*' shall be obtained by applying the 6-tap filter horizontally in an equivalent manner to '*b*' and intermediate values denoted as '*cc*', '*dd*', '*ee*', '*m*' and '*ff*' shall be obtained by applying the 6-tap filter vertically in an equivalent manner to '*h*'. The final prediction value shall be calculated using: j=Clip1((*j*+512)>>10).

– The samples at quarter sample positions labelled as 'a', 'c', 'd', 'n', 'f', 'i', 'k' and 'q' shall be obtained by averaging with truncation the two nearest samples at integer and half sample positions using: a=(G+b)>>1, c=(H+b)>>1, d=(G+h)>>1, n=(M+h)>>1, f=(b+j)>>1, i=(h+j)>>1, k=(j+m)>>1 and q=(j+s)>>1.

– The samples at quarter sample positions labelled as 'e', 'g' and 'p' shall be obtained by averaging with truncation the two nearest samples at half sample positions in the diagonal direction using e=(b+h)>>1, g=(b+m)>>1, p=(h+s)>>1.

– The sample at quarter sample position labelled as 'r' shall be obtained by averaging with rounding using the four nearest samples at integer positions using r=(G+H+M+N+2)>>2.

8.4.2.2 One eighth sample luma interpolation

The positions labelled 'A' in Figure 8-7 represent reference picture samples in integer positions. Other symbols represent interpolated values at fractional sample positions.

A	d	b^h	d	b^h	d	b^h	d	A
d	e	d	f^h	d	f^h	d	e	
b^v	d	c^q	d	c^q	d	c^q	d	b^v
d	f^v	f	g	d	g	d	f^v	
b^v	d	c^q	d	c^m	d	c^q	d	b^v
d	f^v	d	g	d	g	d	f^v	
b^v	d	c^q	d	c^q	d	c^q	r	b^v
d	e	d	f^h	d	f^h	d	e	
A		b^h		b^h		b^h		A

Figure 8-7 – Integer samples ('A') and fractional sample locations for one eighth sample luma interpolation

The samples at half and quarter sample positions shall be obtained by applying 8-tap filters with following coefficients:

- coeff1 for sample values at 1/4 positions: (-3, 12, -37, 229, 71, -21, 6, -1),
- coeff2 for sample values at 2/4 positions: (-3, 12, -39, 158, 158, -39, 12, -3),
- coeff3 for sample values at 3/4 positions: (-1, 6, -21, 71, 229, -37, 12, -3).

The samples at one eighth sample positions are defined as weighted averages of reference picture samples at integer, half and quarter sample positions. The process for each position is described below.

- The samples at half and quarter sample positions denoted as 'b^h' shall be obtained by first calculating intermediate values b, by applying 8-tap filter to the nearest samples 'A' at integer positions in a horizontal direction. For left 'b^h', middle 'b^h' and right 'b^h' in Figure 8-7, coefficients coeff1, coeff2 and coeff3 are used, respectively. The final value of 'b^h' shall be obtained using $b^h = \mathrm{Clip1}((b+128)>>8)$. The samples at half and quarter sample positions labelled as 'b^v' shall be obtained equivalently with the filter applied in vertical direction. For upper 'b^v', middle 'b^v' and bottom 'b^v' coefficients coeff1, coeff2 and coeff3 are used, respectively.
- The samples at half and quarter sample positions labelled as 'c^m' and 'c^q' shall be obtained by 8-tap filtering of the closest intermediate values b in either horizontal or vertical direction to obtain value c, and then the final result shall be obtained using $c^m=\mathrm{Clip1}((c+32768)>>16)$ or $c^q = \mathrm{Clip1}((c+32768)>>16)$. Filtering in horizontal and vertical direction gives identical results. When filtering in horizontal direction is applied, for left 'c^q', middle 'c^q' and right 'c^q', coefficients coeff1, coeff2 and coeff3 are used, respectively. When filtering in vertical direction is applied, for upper 'c^q', middle 'c^q' and bottom 'c^q' coefficients coeff1, coeff2 and coeff3 are used, respectively. For 'c^m' coefficients coeff2 are used.
- The samples at one eighth sample positions labelled as 'd' shall be obtained by averaging with truncation of the two closest samples at half and quarter sample positions using $d = (A+b^h)>>1$, $d = (b^h+b^h)>>1$, $d = (A+b^v)>>1$, $d = (b^h+c^q)>>1$, $d = (b^v-c^q)>>1$, $d = (c^q+c^q)>>1$, $d = (b^v+b^v)>>1$, or $d = (c^q+c^m)>>1$.
- The samples at one eighth sample positions labelled as 'e' shall be obtained by averaging with truncation the closest 'b^h' and 'b^v' samples in diagonal direction using $e = (b^h+b^v)>>1$.
- The samples at one eighth sample positions labelled as 'g' shall be obtained from the closest integer samples 'A' and the 'c^m' samples using $g = (A+3c^m-2)>>2$.
- The samples at one eighth sample positions labelled as 'f^h' and 'f^v' shall be calculated as $f^h - (3b^h+ b^v+2)>>2$ and $f^v = (3b^v+ b^h+2)>>2$.

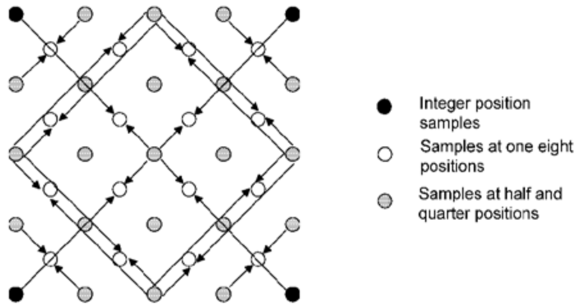

Figure 8-8 – Diagonal interpolation for one eighth sample luma interpolation

8.4.2.3 Chroma interpolation

Motion compensated prediction fractional chroma samples shall be obtained using Equation 8-13.

$$v = ((s-d^x)(s-d^y)A + d^x(s-d^y)B + (s-d^x)d^yC + d^xd^yD + s^2/2)/s^2 \qquad (8\text{-}13)$$

where A, B, C and D are the integer position reference picture samples surrounding the fractional sample location; d^x and d^y are the fractional parts of the sample position in units of one eighth samples for quarter sample interpolation or one sixteenth samples for one eighth sample interpolation; and s is 8 for quarter sample interpolation and is 16 for one eighth sample interpolation. The relationships between the variables in Equation 8-13 and reference picture positions are illustrated in Figure 8-9.

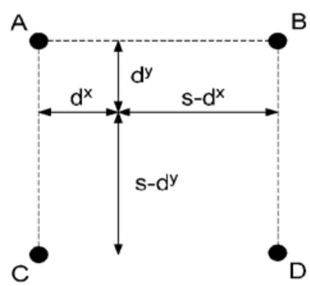

Figure 8-9 – Fractional sample position dependent variables in chroma interpolation and surrounding integer position samples A, B, C, and D.

8.5 Intra Prediction

Two Intra coding modes for macroblocks are described below. Sample values used in the prediction process for intra sample prediction shall be sample values prior to alteration by any deblocking filter operations.

8.5.1 Intra Prediction for 4x4 luma block in Intra_4x4 macroblock type

Figure 8-10 illustrates the Intra prediction for a 4x4 block. The samples of a 4x4 block containing samples a to p in Figure 8-10 are predicted using samples A to Q in Figure 8-10 from neighbouring blocks. Samples A to Q may already be decoded and may be used for prediction. Any sample A-Q shall be considered not available under the following circumstances:

- if they are outside the picture or outside the current slice,
- if they belong to a macroblock that is subsequent to the current macroblock in raster scan order,

- if they are sent later than the current 4x4 block in the order shown in Figure 6-5, or

  if they are in a non-intra macroblock and constrained_intra_pred is 1.

When samples E-H are not available, the sample value of D is substituted for samples E-H. When samples M-P are not available, the sample value of L is substituted for samples M-P.

Q	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				
M								
N								
O								
P								

Figure 8-10 – Identification of samples used for intra spatial prediction

For the luma signal, there are nine intra prediction modes labelled 0, 1, 3, 4, 5, 6, 7, and 8. Mode 2 is 'DC-prediction' (see below). The other modes represent directions of predictions as indicated in Figure 8-11.

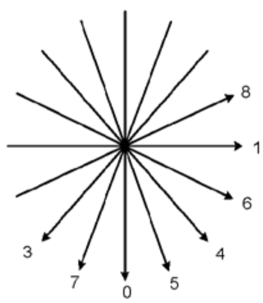

Figure 8-11 – Intra prediction directions

If adaptive_block_size_transform_flag == 1, the intra prediction modes may be used for 4x8, 8x4, and 8x8 blocks as specified in subclause 12.~~3~~4.1.

8.5.1.1 Mode 0: vertical Prediction

This mode shall be used only if A, B, C, D are available. The prediction in this mode shall be as follows:

- a, e, i, m are predicted by A,
- b, f, j, n are predicted by B,
- c, g, k, o are predicted by C,
- d, h, l, p are predicted by D.

8.5.1.2 Mode 1: horizontal prediction

This mode shall be used only if I, J, K, L are available. The prediction in this mode shall be as follows:

– a, b, c, d are predicted by I,
  e, f, g, h are predicted by J,
– i, j, k, l are predicted by K,
– m, n, o, p are predicted by L.

8.5.1.3 Mode 2: DC prediction

If all samples A, B, C, D, I, J, K, L, are available, all samples are predicted by (A+B+C+D+I+J+K+L+4)>>3. If A, B, C, and D are not available and I, J, K, and L are available, all samples shall be predicted by (I+J+K+L+2)>>2. If I, J, K, and L are not available and A, B, C, and D are available, all samples shall be predicted by (A+B+C+D+2)>>2. If all eight samples are not available, the prediction for all luma samples in the 4x4 block shall be 128. A block may therefore always be predicted in this mode.

8.5.1.4 Mode 3: diagonal down/left prediction

This mode shall be used only if all A, B, C, D, I, J, K, L, Q are available. This is a 'diagonal' prediction. The prediction in this mode shall be as follows:

– a is predicted by (A + 2B + C + I + 2J + K + 4) >> 3
– b, e are predicted by (B + 2C + D + J + 2K + L + 4) >> 3
– c, f, i are predicted by (C + 2D + E + K + 2L + M + 4) >> 3
– d, g, j, m are predicted by (D + 2E + F + L + 2M + N + 4) >> 3
– h, k, n are predicted by (E + 2F + G + M + 2N + O + 4) >> 3
– l, o are predicted by (F + 2G + H + N + 2O + P + 4) >> 3
– p is predicted by (G + H + O + P + 2) >> 2

8.5.1.5 Mode 4: diagonal down/right prediction

This mode shall be used only if all A, B, C, D, I, J, K, L, Q are available. This is a 'diagonal' prediction. The prediction in this mode shall be as follows:

– m is predicted by: (J + 2K + L + 2) >> 2
– i, n are predicted by (I + 2J + K + 2) >> 2
– e, j, o are predicted by (Q + 2I + J + 2) >> 2
– a, f, k, p are predicted by (A + 2Q + I + 2) >> 2
– b, g, l are predicted by (Q + 2A + B + 2) >> 2
– c, h are predicted by (A + 2B + C + 2) >> 2
– d is predicted by (B + 2C + D + 2) >> 2

8.5.1.6 Mode 5: vertical-left prediction

This mode shall be used only if all A, B, C, D, I, J, K, L, Q are inside the slice. This is a 'diagonal' prediction.

– a, j are predicted by (Q + A + 1) >> 1
– b, k are predicted by (A + B + 1) >> 1
– c, l are predicted by (B + C + 1) >> 1
– d is predicted by (C + D + 1) >> 1
– e, n are predicted by (I + 2Q + A + 2) >> 2
– f, o are predicted by (Q + 2A + B + 2) >> 2
– g, p are predicted by (A + 2B + C + 2) >> 2
– h is predicted by (B + 2C + D + 2) >> 2
– i is predicted by (Q + 2I + J + 2) >> 2
– m is predicted by (I + 2J + K + 2) >> 2

8.5.1.7 Mode 6: horizontal-down prediction

This mode shall be used only if all A, B, C, D, I, J, K, L, Q are available. This is a 'diagonal' prediction. The prediction in this mode shall be as follows:

– a, g are predicted by (Q + I + 1) >> 1

- b, h are predicted by (I + 2Q + A+ 2) >> 2
- c is predicted by (Q + 2A + B+ 2) >> 2
- d is predicted by (A + 2B + C− 2) >> 2
- e, k are predicted by (I + J + 1) >> 1
- f, l are predicted by (Q + 2I + J+ 2) >> 2
- i, o are predicted by (J + K + 1) >> 1
- j, p are predicted by (I + 2J − K+ 2) >> 2
- m is predicted by (K + L + 1) >> 1
- n is predicted by (J + 2K + L + 2) >> 2

8.5.1.8 Mode 7: vertical-right prediction

This mode shall be used only if all A, B, C, D, I, J, K, L, Q are available. This is a 'diagonal' prediction. The prediction in this mode shall be as follows:

- a is predicted by (2A + 2B + J + 2K + L + 4) >> 3
- b, i are predicted by (B + C + 1) >> 1
- c, j are predicted by (C + D + 1) >> 1
- d, k are predicted by (D + E + 1) >> 1
- l is predicted by (E + F + 1) >> 1
- e is predicted by (A + 2B + C + K + 2L + M + 4) >> 3
- f, m are predicted by (B + 2C + D + 2) >> 2
- g, n are predicted by (C + 2D + E + 2) >> 2
- h, o are predicted by (D + 2E + F + 2) >> 2
- p is predicted by (E + 2F + G + 2) >> 2

8.5.1.9 Mode 8: horizontal-up prediction

This mode shall be used only if all A, B, C, D, I, J, K, L, Q are available. This is a 'diagonal' prediction. The prediction in this mode shall be as follows:

- a is predicted by (B + 2C + D + 2I + 2J + 4) >> 3
- b is predicted by (C + 2D + E + I + 2J + K + 4) >> 3
- c, e are predicted by (J + K + 1) >> 1
- d, f are predicted by (J + 2K + L + 2) >> 2
- g, i are predicted by (K + L + 1) >> 1
- h, j are predicted by (K + 2L + M + 2) >> 2
- l, n are predicted by (L + 2M + N + 2) >> 2
- k, m are predicted by (L + M + 1) >> 1
- o is predicted by (M + N + 1) >> 1
- p is predicted by (M + 2N + O + 2) >> 2

8.5.2 Intra prediction for luma block in Intra_16x16 macroblock type

Denote the block to be predicted as having sample locations 0 to 15 horizontally and 0 to 15 vertically. The notation P(x,y) is used, where x = 0..15 corresponds to horizontal positions and y = 0..15 corresponds to vertical positions. P(x,-1), x=0..15 are the neighbouring samples above the block and P(-1,y), y=0..15 are the neighbouring samples to the left of the block. Pred(x,y) x,y = 0..15 is the prediction for the luma macroblock samples. There are 4 different prediction modes as specified in subclauses 8.5.2.1 to 8.5.2.4.

Samples P(x,-1) or P(-1,y) shall be considered not available under the following circumstances:

- if they are outside the picture or outside the current slice, or
- if they are in a non-intra macroblock and constrained_intra_pred is 1.

8.5.2.1 Mode 0: vertical prediction

This mode shall be used only if all neighbouring samples P(x, -1) are available.

$$\text{Pred}(x, y) = P(x, -1),\ x, y=0..15 \tag{8-14}$$

8.5.2.2 Mode 1: horizontal prediction

This mode shall be used only if all neighbouring samples P(-1, y) are available.

$$\text{Pred}(x, y) = P(-1, y),\ x, y=0..15 \tag{8-15}$$

8.5.2.3 Mode 2: DC prediction

$$\text{Pred}(x, y) = [\sum_{x'=0}^{15} P(x',-1) + \sum_{y'=0}^{15} P(-1,y') + 16] >> 5 \quad x, y=0..15 \tag{8-16}$$

If the neighbouring samples P(x, -1) are not available and the neighbouring samples P(-1, y) are available, the prediction for all luma samples in the macroblock is given by Equation 8-17.

$$\text{Pred}(x, y) = [\sum_{y'=0}^{15} P(-1,y') + 8] >> 4 \quad x, y=0..15, \tag{8-17}$$

If the neighbouring samples P(-1, y) are not available and the neighbouring samples P(x, -1) are available, the prediction for all luma samples in the macroblock is given by Equation 8-18.

$$\text{Pred}(x, y) = [\sum_{x'=0}^{15} P(x',-1) + 8] >> 4 \quad x, y=0..15, \tag{8-18}$$

If none of the neighbouring samples P(x,-1) and P(-1,y) are available, the prediction for all luma samples in the macroblock shall be 128.

8.5.2.4 Mode 3: plane prediction

This mode shall be used only if all neighbouring samples P(x, -1) and P(-1, y) are available.

$$\text{Pred}(x,y) = \text{Clip1}(\ (a + b\cdot(x-7) + c\cdot(y-7) +16) >> 5\ ), \tag{8-19}$$

where:

$$a = 16\cdot(P(-1,15) + P(15,-1)) \tag{8-20}$$

$$b = (5*H+32)>>6 \tag{8-21}$$

$$c = (5*V+32)>>6 \tag{8-22}$$

and H and V are defined in Equations 8-23 and 8-24.

$$H = \sum_{x=1}^{8} x \cdot (P(7+x,-1) - P(7-x,-1)) \tag{8-23}$$

$$V = \sum_{y=1}^{8} y \cdot (P(-1,7+y) - P(-1,7-y)) \tag{8-24}$$

8.5.3 Prediction in intra coding of chroma blocks

The chroma in intra macroblocks is predicted in a manner very similar to the luma block in Intra_16x16 macroblock type (subclause 8.5.2), using one of four prediction modes. The same prediction mode is applied to both chroma blocks, but it is independent of the prediction mode used for the luma.

NOTE - If any portion of the luma macroblock is coded in intra mode, the entire chroma macroblock is coded intra.

Let P(x,-1), x=0..7 be the neighbouring samples above the chroma macroblock and P(-1,y), y=0..7 be the neighbouring samples to the left of the chroma macroblock. Pred(x,y), x,y = 0..7 is the prediction for the whole chroma macroblock,

and is computed as follows for the four prediction modes. Samples P(x,-1) or P(-1,y) shall be considered not available under the following circumstances:

- if they are outside the picture or outside the current slice,
- if they are in a non-intra macroblock and constrained_intra_pred is 1.

Whenever P(x,y) is ~~unavailable~~not available, P(x,y) is ~~set to~~inferred to have the value 128 except as specified in subclause 8.5.3.3.

For the horizontal and vertical prediction, P(x,y) is first filtered using a {1,2,1}/4 filter, with pixel replication at the edges.

8.5.3.1 Mode 0: vertical prediction

$$F(0,-1) = (P(0,-1) + P(1,-1) + 1) >> 1 \quad (8\text{-}25)$$

Formatted: Lowered by 5 pt

$$F(x,-1) = (P(x-1,-1) + 2 \times P(x,-1) + P(x+1,-1) + 2) >> 2,\ x=1,\ldots,6 \quad (8\text{-}26)$$

Formatted: Lowered by 5 pt

$$F(7,-1) = (P(6,-1) + P(7,-1) + 1) >> 1 \quad (8\text{-}27)$$

$$\text{Pred}(x, y) = F(x, -1),\ x, y=0..7 \quad (8\text{-}28)$$

8.5.3.2 Mode 1: horizontal prediction

$$F(-1,0) = (P(-1,0) + P(-1,1) + 1) >> 1 \quad (8\text{-}29)$$

Formatted: Lowered by 5 pt

$$F(-1,y) = (P(-1,y-1) + 2 \times P(-1,y) + P(-1,y+1) + 2) >> 2,\ y=1,\ldots,6 \quad (8\text{-}30)$$

$$F(-1,7) = (P(-1,6) + P(-1,7) + 1) >> 1 \quad (8\text{-}31)$$

$$\text{Pred}(x, y) = F(-1, y),\ x, y=0..7 \quad (8\text{-}32)$$

8.5.3.3 Mode 2: DC prediction

If all samples P(-1,n) and P(n,-1) used in Equation 8-33 are available, the prediction is formed as

Formatted: Normal

$$\text{Pred}(x, y) = \left(\left(\sum_{n=0}^{7}(P(-1,n) + P(n,-1))\right) + 8\right) >> 4 \quad x,y=0..7, \quad (8\text{-}33)$$

If the 8 samples P(-1,n) are not available, the prediction is formed as

Formatted: Font: Italic

$$\text{Pred}(x,y) = \left[\left(\sum_{n=0}^{7} P(n,-1)\right) + 4\right] >> 3 \quad x,y=0..7, \quad (8\text{-}33a)$$

If the 8 samples P(n,-1) are not available, the prediction is formed as

Formatted: Lowered by 16 pt

$$\text{Pred}(x,y) = \left[\left(\sum_{n=0}^{7} P(n,-1)\right) + 4\right] >> 3 \quad x,y=0..7, \quad (8\text{-}33b)$$

~~where only the average of 8 samples is used when the other 8 samples are unavailable.~~ If all 16 samples are ~~unavailable~~not available, the prediction Pred(x,y) for all samples x,y=0..7~~in the macroblock~~ is 128.

8.5.3.4 Mode 3: plane prediction

For the plane mode, ~~we adapt the luma 16x16 plane mode coefficients to an 8x8 block,~~ the prediction is formed as ~~follows~~:

$$\text{Pred}(i,j) = \text{Clip1}(\ (a + b\cdot(i\text{-}3) + c\cdot(j\text{-}3) + 16) >> 5\ ),\ i,j=0,\ldots,7 \quad (8\text{-}34)$$

where: